JHN

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/027,353	12/19/2001	Walter T. Nixon	5681-05300	9647

7590 03/28/2005

Robert C. Kowert
Conley, Rose & Tayon, P.C.
P.O. Box 398
Austin, TX 78767

EXAMINER	
TSAI, SHENG JEN	

ART UNIT	PAPER NUMBER
2186	

DATE MAILED: 03/28/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/027,353	NIXON ET AL.	
	Examiner	Art Unit	
	Sheng-Jen Tsai	2186	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 March 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-36* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-36* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *03/11/2005*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

re

DETAILED ACTION

1. This Office Action is taken in response to Applicants' Amendment and Remarks filed on March 11, 2005 regarding application 10/027,353 filed on December 19, 2001. The examiner acknowledges receiving of additional Information Disclosure Statement filed on March 11, 2005.

2. Claims 1-36 are pending in the application under consideration.

Original claims 1-11, 17-22 and 33-35 have been amended.

3. ***Response to Amendment and Remarks***

Applicant's amendments and remarks have been fully and carefully considered with the results detailed below.

As to amendment for claims 1-11, 17-22 and 33-35:

In these claims the term "cache accumulator" has been amended to be "cache accumulator memory" and to be consistent with description in the specification section.

As such, the cause that leads to the 35 USC 112, second paragraph rejection stated in the previous Office Action no longer exists and the rejection is nullified.

As to amendment and remarks for claims 1, 20, and 33-34:

Independent claims 1, 20, and 33-34 are currently amended with the additional limitation of "**wherein the cache accumulator memory is configured as a cache of the memory**."

However, the prior art cited in the previous Office Action (Fossum et al., US 4,888,679) clearly illustrates that the cache memory (figure 1, 24; figure 2, 24; figure 7,

106) is configured as a cache of the main memory (figure 1, 23; figure 2, 23; figure 7, 101). Therefore this additional limitation is anticipated by Fossum et al..

Independent claim 1 is currently further amended with the additional limitation of "**wherein the cache accumulator memory is configured to accumulate an intermediate result of the first accumulation operation, wherein the intermediate result is both a result of and an operand of the first accumulation operation**;"

While Fossum et al. do not explicitly state that the cache memory is configured to accumulate an intermediate result which serves as both an operand and a result, it is noted that this is exactly the intended purpose of a cache memory. It is well-known and understood by one skilled in the art that a cache memory is used to store data that is needed currently, or soon to be needed, by the processor to increase the "hit" ratio and hence reduce access latency. Thus intermediate results will certainly be accumulated and stored in the cache memory, rather than in the main memory, to take advantage of the higher speed of the cache memory. Figure 7 clearly shows that the cache memory, being coupled between the main memory and the function units (vector processor and scalar processor) is configured to store, accumulate, and provide intermediate results.

Furthermore, in response to this newly amended limitation, a new prior art (Sollars, US 6,216,218) has been identified. Sollars explicitly disclose in the invention "Processor Having a Datapath and Control Logic Constituted with Basis Execution Blocks" an apparatus comprising a cache memory and ALU selected coupled to each other, to allow operands to be directly supplied to ALU and stored back into cache memory (figure 7; figure 13; column 7, lines 12-17; column 4, lines 9-36). Sollars'

invention provides another instance of evidence that a cache memory is intended for storing intermediate results and for supplying operands for next operation. Therefore it would have been obvious for persons of ordinary skills in the art at the time of applicant's invention to recognize the common practice of using a cache memory for storing intermediate results and for supplying operands for next operation, as implied by Fossum et al. and explicitly illustrated by Sollars in their respective invention, and to anticipate it as such in their own invention.

Therefore, the examiner's position regarding these claims, and those claims dependent from them, remain the same as stated in the previous Office Action. However, the grounds of rejections for independent claims 1, 20, and 33-34 are now based on 35 USC 103(a) instead of 102(b).

4. ***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1, 4-6, 9-12, and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fossum et al. (U.S. 4,888,679), and in view of Sollars (US 6,216,218).

As to claim 1, Fossum et al. disclose **an apparatus comprising:**

A memory [figure1 shows a main memory unit (item 23)];

A functional unit [figure 1 shows a vector processor unit (item 22) and a scalar processor (item 21), both are functional units] **configured to perform a block operation on one or more block operands to generate a block result** [a cache bypass is provided to transmit data directly to the vector processor as the data from the main memory are being stored in the cache (column 3, lines 1-4); the vector processor prefetch requests include the virtual address of the blocks that will be accessed by the vector processor (column 3, lines 20-22); Figure 2 shows that how blocks of data are stored in the main memory; and figure 7 shows that the vector processing unit having ADD, MASK and MULTIPLY sub-units for operating on the blocks of data]; **and**

a cache accumulator memory coupled to the memory and the functional unit [figure 1 shows that a cache memory (item 24) is coupled to the main memory (item 23) and the functional unit (item 22, the vector processor and item 21, the scalar processor)], **wherein the cache accumulator memory comprises a plurality of block storage locations** [figure 2 shows that the cache memory (item 24) has a plurality of block storage locations (item 33, data storage)], **wherein the cache accumulator memory is configured to receive a set of one or more instructions to perform a first accumulation operation** [figure 7 shows that the cache memory (item 106) is configured to receive a set of one or more instructions from an instruction processing unit (item 107) to support the executions of instructions], **wherein a first instruction in the set uses a first address in the memory to identify a first block operand** [in response to a prefetch request, the cache is checked to determine whether it includes the required block, and if the cache does not have the required

block, a refill request is sent to the main memory (column 2, lines 53-57); the vector processor receives a vector load instruction which commands the vector processor to send vector element addresses to the cache memory, and in response to those element addresses, the cache transmits the desired vector elements to the vector processor (column 4, lines 57-62)]; **wherein in response to receiving the first instruction in the set, the cache accumulator memory is configured to access an associativity list comprising an indication that a first set of the block storage locations is allocated to the first accumulation operation and, in response to the indication, to provide the first block operand to the functional unit from the first set of block storage locations** [The cache includes means for storing selected predefined blocks of data elements, means for receiving requests from the scalar processor to access a specified data element, means for checking whether the data element is in a block stored in the cache, and means operative when data for the block including the specified data element is not so stored for reading the specified block of data from the main memory and storing the block of data in the cache (column 4, lines 25-33); If a data element needed by the scalar processor is not found in the cache, then the data element is obtained from the main memory, but in the process an entire block, including additional data, is obtained from the main memory and written into the cache. Due to the principle of locality in time and memory space, the subsequent times that the scalar processor 21 desires a data element, chances are that this data element will be found in the block which includes the previously addressed data element. Therefore, chances are that the cache will already include the data element desired by

the scalar processor (column 4, lines 36-47)] **and to store the block result generated by the functional unit into the first set of block storage locations** [figure 7 shows that the block data generated by both the vector processor (item 116) and the scalar processor (item 108) are piped through the register file & arithmetic logic unit (item 111) and the cache bypass mux unit (item 135) to return to the data storage (item 114) of the cache unit (item 106)].

Further, refer to "**As to amendment and remarks for claims 1, 20, and 33-34**" for explanation regarding newly added limitations.

As to claim 4, Fossum et al. do not explicitly mention that **the cache accumulator is configured to indicate whether a particular block operand stored in the cache accumulator is modified with respect to a copy of that particular block operand in the memory**, since the disclosure focuses on the aspect of vector processing using a data cache. However, it is inherent for all cache memory systems that a mechanism is required to maintain the consistency between the main memory and the cache memory, and as such an indicator, commonly known as the "dirty bit," is required to indicate whether the data in the cache has been modified and hence is different from the corresponding copy in the main memory. Therefore, this claim is anticipated by the invention of Fossum et al.

As to claim 5, Fossum et al. disclose that the cache includes means for storing selected predefined blocks of data elements, means for receiving requests from the scalar processor to access a specified data element, means for checking whether the data element is in a block stored in the cache, and means operative when data for the

block including the specified data element is not so stored for reading the specified block of data from the main memory and storing the block of data in the cache (column 4, lines 25-33); and that If a data element needed by the scalar processor is not found in the cache, then the data element is obtained from the main memory (column 4, lines 36-37).

As to claim 6, Fossum et al. disclose that the cache includes an input address register generally designated, a tag store generally designated, and a data store generally designated. The data store is organized for storing selected ones of the predefined blocks of the data elements. In order to indicate whether data for a specified block are stored in the data store, the tag store is organized for storing respective tags associated with the blocks (column 6, lines 30-38). Also, as specifically shown in figure 2, the tag comprises the upper portion of the block address. In response to a fill request, an addressed block in the main memory is transferred to one or more predefined slots in the data store. The slots associated with a given block are indexed by an index j. The index j and the tag for a particular block specify the block address for that block. Therefore, when an address of a desired byte is received in the input register, the index portion j points to at least one corresponding slot in the tag store and the addressed tag is fed to the comparator for comparison with the tag specified by the byte address (column 6, lines 41-52). Hence, the use of the tags as described fulfills the functions recited in applicant's claim 6.

As to claim 9, Fossum et al. disclose that the cache includes an input address register generally designated, a tag store generally designated, and a data store

generally designated. The data store is organized for storing selected ones of the predefined blocks of the data elements. In order to indicate whether data for a specified block are stored in the data store, the tag store is organized for storing respective tags associated with the blocks (column 6, lines 30-38; figure 2, items 31 and 32)). Also, as specifically shown in figure 2, the tag comprises the upper portion of the block address. In response to a fill request, an addressed block in the main memory is transferred to one or more predefined slots in the data store. The slots associated with a given block are indexed by an index j. The index j and the tag for a particular block specify the block address for that block. Therefore, when an address of a desired byte is received in the input register, the index portion j points to at least one corresponding slot in the tag store and the addressed tag is fed to the comparator for comparison with the tag specified by the byte address (column 6, lines 41-52). Hence, the use of the tags as described fulfills the functions recited in applicant's claim 9.

As to claim 10, the applicant describes the aspect of "write into the cache from the functional unit," while the scenario described in claim 9 is the aspect of "write into the cache from the main memory." As far as Fossum et al.'s invention is concerned, both cases have the same impacts on the cache of modifying the contents of a block location within the cache, therefore the same technique of updating the tags associated with the involved block storage locations to indicate that data is stored within the associated locations is equally applicable. Refer to the claim analysis provided in "As to claim 9."

As to claim 11, Fossum et al. disclose that (1) a tag store is organized for storing respective tags associated with the block storage locations to indicate whether data for a specified block are present in the data store (column 6, lines 35-38), (2) the tag field occupies the first portion of the byte address bits (figure 2, item 31), and (3) the results from the functional unit (the vector processor) are routed back into the cache. Thus if the instruction (for example, an ADD instruction of A+B=C) is such that it takes two operands from two different locations (A and B) and stores the sum to a third location (C) then it will cause the tag associated with location C to be modified to indicate the presence of new data of C at that location. This effectively sets the first portion of a second address (i.e., the address of C) in the memory to identify the block result.

As to claim 12, as explained in "As to claim 11" using the example of an ADD instruction of A+B=C, the address of the result (the second address) is different to that of the operands (the first address).

As to claim 16, Fossum et al. disclose that **the functional unit** [the vector processing unit shown in figure 7, item 116] **is configured to perform the operation on two block operands** [figure 7 shows that the vector processing unit performs ADD and MULTIPLY operations. Both ADD and MULTIPLY operations require two operands].

6. Claims 2 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fossum et al. (U.S. 4,888,679), in view of Sollars (US 6,216,218), and further in view of McClure (U.S. 5,590,307).

As to claim 2, Fossum et al. and Sollars do not mention that **the cache accumulator memory comprises a dual-ported memory**. However, McClure explicitly discloses the invention of a dual-port data cache memory having one port dedicated to serving a local processor and a second port dedicated to serving a system (abstract, figure 2). A dual-port cache memory allows data to be transferred between the cache and other entities of the system, such as the main memory, at a higher speed as compared to a one-port cache memory. Since data transfer to and from the cache is unavoidable when a miss occurs, a higher data transfer speed will reduce the memory latency and improve the throughput of the system. Therefore it would have been obvious for persons of ordinary skills in the art at the time of applicant's invention to recognize the benefits offered by a dual-port cache memory and to use it as the cache unit in the apparatus disclosed by Fossum et al. to further improve its performance.

As to claim 19, Fossum et al. and Sollars do not mention that **the cache accumulator memory is configured to store a word of the block result during an access cycle in which cache accumulator also provides a word of the first block operand to the means for performing a block operation**. However, McClure discloses the invention of a dual-port data cache memory having one port dedicated to serving a local processor and a second port dedicated to serving a system (abstract; figure 2). A dual-port cache memory allows two pieces of data to be accessed, one at each port, at the cache by other entity of the system at the same time, hence enabling the cache to serve the functional block with two pieces of data simultaneously (figure 2). In other words, the cache will be able to store a word of the block result from the

functional unit (a write operation into the cache), and during the same access cycle provides a word to the functional unit (a read operation from the cache). This type of concurrent operations increases the system throughput. Therefore it would have been obvious for persons of ordinary skills in the art at the time of applicant's invention to recognize the benefits offered by a dual-port cache memory in supporting concurrent operations, and to use it as the cache unit in the apparatus disclosed by Fossum et al. to further improve its performance.

7. Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over Fossum et al. (U.S. 4,888,679), in view of Sollars (US 6,216,218), and further in view of Faraboschi et al. (U.S. 6,122,708).

As to claim 3, Fossum et al. and Sollars do not mention that **the cache accumulator memory comprises at least two independently interfaced memory banks,** although Fossum et al. do disclose the use of a plurality of main memory banks (figure 2, item 23) and teach that block interleaving of the memory bank addresses is useful in practicing the disclosed invention since vectors are stored and referenced in a linear fashion with respect to the physical addresses of the bytes in the main memory. When a vector extends across one or more block boundaries, it is desirable for multiple ones of the contiguous blocks to be simultaneously accessed in the main memory using multiple banks (column 6, lines 6-13). Further, Faraboschi et al. discloses a data cache system for use with streaming data in which the data cache consists of two independently interfaced memory banks (figure 3, items 130 and 132), that The data cache memory may include a single bank, or two or more banks in a set associative

configuration, with each bank includes a data cache, a tag array, and addressing circuitry (column 3, lines 47-50). Two-bank organization of the cache system allows data to be transferred to and from the cache system simultaneously using the two banks, such as **providing the first block operand from a first storage location in a first one of the independently interfaced memory banks and to store the block result in a second block storage location in a second one of the independently interfaced memory banks, wherein the first set of block storage locations comprises the first block storage location and the second block storage location** (this is the case where a vector/block extends across one or more block boundaries explained earlier), hence avoiding the situation where a single-bank cache becomes the bottleneck of memory access and will reduce the overall memory access latency. Therefore it would have been obvious for persons of ordinary skills in the art at the time of applicant's invention to recognize the benefits offered by a two-bank cache memory architecture and to adopt it for the cache unit in the apparatus disclosed by Fossum et al. to further improve its performance.

8. Claims 7 and 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fossum et al. (U.S. 4,888,679), in view of Sollars (US 6,216,218), and further in view of Handy, "The cache memory book: the authoritative reference on cache design," Academic Press, 1993, page 57.

As to claim 7, Fossum et al. and Sollars do not explicitly mention that **the cache accumulator is configured to use a least recently used algorithm to select the first set of block storage locations to overwrite**, since the disclosure focuses on the

aspect of vector processing using a data cache. However, Handy teaches that a replacement algorithm is required in a cache system to select which entry in the cache is to be replaced when a new line is to be brought into the cache, and that the least recent used algorithm is one of the most commonly adopted scheme. Therefore it would have been obvious for persons of ordinary skills in the art at the time of applicant's invention to recognize the need to have a replacement algorithm and the benefit offered by a least recently used algorithm and to adopt it for the cache unit in the apparatus disclosed by Fossum et al.

As to claim 8, Fossum et al. and Sollars do not explicitly mention that **if data to be overwritten in the first set of block storage locations is modified with respect to a copy of that data in the memory, the cache accumulator is configured to write the data back to the memory before loading the copy of the first block operand into the first set of block storage locations**, since the disclosure focuses on the aspect of vector processing using a data cache. However, Handy teaches that a write strategy is required in a cache system to deal with the situations where data is modified in either he cache or the main memory, which leads to data inconsistency between the main memory and cache. Particularly, Handy teaches that a technique, known as "write –through," in which the main memory is always updated first during all write cycles, is commonly adopted in cache system design (pages 64-65). With such a write-through policy, data consistency between the main memory and the cache will be enforced. Therefore it would have been obvious for persons of ordinary skills in the art at the time of applicant's invention to recognize the need to have a write policy and the

benefit offered by the write-through algorithm, and to adopt it for the cache unit in the apparatus disclosed by Fossum et al.

9. Claims 13, 14, and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fossum et al. (U.S. 4,888,679), in view of Sollars (US 6,216,218), and further in view of "Microsoft Computer Dictionary," Microsoft Press, 2002, page 391: parity.

As to claim 13, Fossum et al. and Sollars do not mention that **the functional unit is configured to perform a parity calculation on the block operand**. However, parity is a well-known technique in the art and is commonly used for error checking/correction of data transmitted between a source and a destination to ensure the data integrity. Therefore it would have been obvious for persons of ordinary skills in the art at the time of applicant's invention to recognize the benefits offered by the parity scheme for data integrity and to adopt it for the functional unit in the apparatus disclosed by Fossum et al. so that applications with heavy data traffic (there is a lot of data traffic between the cache memory and the main memory as well as the functional unit) are processed with high data reliability.

As to claim 14, refer to the claim analysis provided in "As to claim 13." Further, figure 7 shows the instructions/commands are provided by an instruction process unit (item 107).

As to claim 15, Microsoft Computer Dictionary specifically points out that **the parity may be calculated from a plurality of blocks in a stripe of data** [if checked on a block-by-block basis, the method is called longitudinal redundancy checking, or

LRC (page 391)]. Further, Fossum et al. show in figure 2 that **the data to be processed is organized as a plurality of blocks** (BLOCK 0, BLOCK 1, BLOCK 2, etc. as items 26, 27 and 28 in figure 2) **which are distributed among a number of memory banks in a stripe of data** (figure 2, item 23); and since the functional unit disclosed by Fossum et al. includes a vector processor, which lends itself to operate on blocks of data very efficiently. As far as the sequence of the blocks is concerned, the blocks of data are to be loaded from the main memory into the cache and then into the vector processor, therefore the order of the data based on which the parity is to be calculated is preserved as the order by which the blocks are delivered from the main memory originally, hence **the first block operand is a first one of the data blocks in the stripe of data**.

10. Claims 17 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fossum et al. (U.S. 4,888,679), in view of Sollars (US 6,216,218), and further in view of Morton (U.S. 6,088,783).

As to claim 17, Fossum et al. and Sollars do not explicitly mention that **the first operand is stored in the cache accumulator memory and the second operand is provided on a data bus coupled to provide operands to the functional unit**. However, Morton discloses a Digital Signal Processing unit having a data cache (figure 1, item 108) and a plurality of functional units (figure 1, item 110~113, Arithmetic Units) and is capable of supporting both scalar and vector operations (figure 9, items 107 and 110/111/112/113). In figure 6, Morton shows that the operands may come form the X-bar Switch (item 619), which is directly connected to the data cache (figure 1, the data

cache is connected to X-bar Switch via links 64b, and then to one of the ALU via another link 64b). Thus one of the operand to the functional unit (ALU) may come from the cache memory. In figure 6, Morton also shows that the second operand may come from an "Immediate Data" bus (item 620). Functional units capable of receiving data from various different sources allows operations requiring different data sources to proceed without waiting for a particular data to arrive, hence improve the throughput of the data processing system. Therefore it would have been obvious for persons of ordinary skills in the art at the time of applicant's invention to recognize the benefits offered by a functional unit with multiple data sources in improving the system throughput and to incorporate it into the apparatus disclosed by Fossum et al. to further enhance the performance.

As to claim 18, Fossum et al. and Sollars do not explicitly mention that **the first operand is stored in the cache accumulator memory and the second operand is provided from the main memory**. However, Morton discloses a Digital Signal Processing unit having a data cache (figure 1, item 108), a main memory (figure 1, items 101 and 116), and a plurality of functional units (figure 1, item 110~113, Arithmetic Units) and is capable of supporting both scalar and vector operations (figure 9, items 107 and 110/111/112/113). In figure 6, Morton shows that the operands may come form the X-bar Switch (item 619), which is directly connected to the data cache (figure 1, the data cache is connected to X-bar Switch via links 64b, and then to one of the ALU via another link 64b). Thus one of the operand to the functional unit (ALU) may come from the cache memory. In figure 6, Morton also shows that the second

operand may come directly from the memory bus (items C, 2C and C/2), thus the second operand may come for the main memory. Functional units capable of receiving data from various different sources allows operations requiring different data sources to proceed without waiting for a particular data to arrive, hence improve the throughput of the data processing system. Therefore it would have been obvious for persons of ordinary skills in the art at the time of applicant's invention to recognize the benefits offered by a functional unit with multiple data sources in improving the system throughput and to incorporate it into the apparatus disclosed by Fossum et al. to further enhance the performance.

Claim Rejections - 35 USC § 103

11. Claims 20, 23, 26-27, and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fossum et al. (U.S. 4,888,679), and in view of Sollars (US 6,216,218).

As to claim 20, Fossum et al. a method of performing a block accumulation operation using a cache accumulation memory that comprises a plurality of block storage locations [refer to claim analysis provided in "As to claim 1"].

As to claim 23, refer to claim analysis provided in "As to claim 6."

As to claim 26, refer to claim analysis provided in "As to claim 9" and "As to claim 10."

As to claim 27, refer to claim analysis provided in "As to claim 11."

As to claim 30, refer to claim analysis provided in "As to claim 16."

12. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Fossum et al. (U.S. 4,888,679), in view of Sollars (US 6,216,218), and further in view of McClure (U.S. 5,590,307).

As to claim 21, refer to claim analysis provided in "As to claim 2."

13. Claim 22 is rejected under 35 U.S.C. 103(a) as being unpatentable over Fossum et al. (U.S. 4,888,679), in view of Sollars (US 6,216,218), and further in view of Faraboschi et al. (U.S. 6,122,708).

As to claim 22, refer to claim analysis provided in "As to claim 3."

14. Claims 24 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fossum et al. (U.S. 4,888,679), in view of Sollars (US 6,216,218), and further in view of Handy, "The cache memory book: the authoritative reference on cache design," Academic Press, 1993, page 57.

As to claim 24, refer to claim analysis provided in "As to claim 7."

As to claim 25, refer to claim analysis provided in "As to claim 8."

15. Claims 28 and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fossum et al. (U.S. 4,888,679), in view of Sollars (US 6,216,218), and further in view of "Microsoft Computer Dictionary," Microsoft Press, 2002, page 391: parity.

As to claim 28, refer to claim analysis provided in "As to claim 13."

As to claim 29, refer to claim analysis provided in "As to claim 14."

16. Claims 31 and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fossum et al. (U.S. 4,888,679), in view of Sollars (US 6,216,218), and further in view of Morton (U.S. 6,088,783).

As to claim 31, refer to claim analysis provided in "As to claim 17."

As to claim 32, refer to claim analysis provided in "As to claim 18."

Claim Rejections - 35 USC § 103

17. Claim 33 is rejected under 35 U.S.C. 103(a) as being unpatentable over Fossum et al. (U.S. 4,888,679), and in view of Sollars (US 6,216,218).

As to claim 33, refer to claim analysis provided in "As to claim 1."

Claim Rejections - 35 USC § 103

18. Claim 34 is rejected under 35 U.S.C. 103(a) as being unpatentable over Fossum et al. (U.S. 4,888,679), and in view of Sollars (US 6,216,218).

As to claim 34, refer to claim analysis provided in "As to claim 1."

Claim Rejections - 35 USC § 103

19. Claims 35 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fossum et al. (U.S. 4,888,679), in view of Sollars (US 6,216,218, in view of Morton (U.S. 6,088,783), and further in view of "Microsoft Computer Dictionary," Microsoft Press, 2002, page 391: parity.

As to claim 35, refer to claim analysis provided in "As to claim 17" and "As to claim 13."

As to claim 36, refer to claim analysis provided in "As to claim 15." Further, Fossum et al. show that the data stored in the main memory is such that successive words (words 0-63 in item 26 of figure 2) constitute the first block of operand (BLOCK 0 of figure 2) located within one bank (item 26, figure 2), and that successive words (words 64-127 in item 27 of figure 2) constitute the second block of operand (BLOCK 1

of figure 2) located within another bank (item 27, figure 2), and so on. Essentially, the plurality of blocks of data is distributed in a stripe of data as shown in figure 2 (items 26, 27, 28, and 29). Hence, when a block of data is loaded from the main memory into the cache, and then from cache to the functional unit where the parity is to be calculated, **the first block operand is a first one of the data blocks in the stripe of data and the second block operand is a second one of the data block in the stripe of data**.

Conclusion

20. Claims 1-36 are rejected as explained above.

21. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

22. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sheng-Jen Tsai whose telephone number is 571-272-4244. The examiner can normally be reached on 8:30 - 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Kim can be reached on 571-272-4182. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Sheng-Jen Tsai
Examiner
Art Unit 2186

March 22, 2005

PIERRE BATAILLE
PRIMARY EXAMINER